

NEW CONCEPTS FOR ELECTRONIC FLIGHT DATA MANAGEMENT IN AIRPORT
TRAFFIC CONTROL TOWERS

By:
Todd R. Truitt, Ph.D.
FAA Human Factors Team – Atlantic City, AJP-6110
William J. Hughes Technical Center
Atlantic City International Airport, NJ 08405
USA
Phone: (609) 485-4351; Fax: (609) 485-6218
Todd.Truitt@faa.gov

PRESENTED FOR THE
2007 WORLDWIDE AIRPORT TECHNOLOGY TRANSFER CONFERENCE
Atlantic City, New Jersey, USA

April 2007

INTRODUCTION

Airport operations logged by the 449 Federal Aviation Administration (FAA) Airport Traffic Control Towers (ATCTs) are projected to increase from 63.1 million in 2004 to 68.8 million in 2008 [1]. In anticipation of the increase in air traffic, the FAA is investigating the potential effects of implementing an electronic flight data system (EFDS) in ATCTs. The EFDS would replace the paper Flight Progress Strips (FPSs) used by controllers since the 1930s. Durso and Manning [2] state that the FPS has become an historical artifact that limits the usefulness of flight data and consumes valuable cognitive resources. In today's ATCT environment, controllers must manually update information, record clearances, and physically pass FPSs from one controller to another. The inherent physical limitations of FPSs also restrict the controllers' ability to communicate flight data both within and between facilities.

We must conduct concept research to explore new ways of handling air traffic and its associated flight data. One approach is to transform the current FPS into an electronic format without changing the fundamental way that work is accomplished. However, there is the risk that without fundamental changes to the ATC task, controller workload may become overwhelming and prove a limiting factor to the overall NAS performance. We must explore new approaches and their risks, while keeping user acceptance in mind.

This report describes the processes and methodologies used to create two prototype Electronic Flight Data Interfaces (EFDIs) for the FAA ATCTs. We limited the scope of the prototype EFDIs to the controllers' most common tasks. A limited scope was necessary to simplify the task and to enable examination of the basic concepts of using Electronic Flight Data (EFD). This report also presents an overview of the prototype EFDIs. Truitt [3] presents detailed descriptions of the EFDIs.

LITERATURE REVIEW

Truitt [4] conducted a literature review to examine prior task analyses, published literature, and field observation data to understand the basic functionality of FPSs and flight data in ATCTs. Surprisingly, researchers have conducted very little work during the past 20 years in the ATCT domain, especially in the United States. The literature review provided insights into the potential risks and benefits associated with EFD. A primary benefit of EFD is the ability to share and transfer important flight data. Other benefits of EFD include the ability to present the right flight data to the controller at the right time while still preserving the ability to access all information about a flight; simplified or automated data input; integration with other systems; reduction of workload associated with flight data management; integration of information sources to reduce visual shifts of attention; increased time for controllers to observe the airport surface; increased awareness of other controllers' actions through linked displays; and automatic recording and reporting of activity, such as number and duration of departure delays. Potential disadvantages of EFD include initial training, increased criticality of data entry, and changes to controller selection and training methods. We incorporated these findings into the present design of two prototype EFDIs.

WORKING GROUP

Following the literature review, we formed a multidisciplinary working group that included ATCT controllers, cognitive psychologists, and software developers. The purpose of the working

group was not to design a Graphical User Interface (GUI). Rather, the working group defined the tasks and information needs of ATCT controllers.

The working group began by defining the scope of the interfaces to be designed. They limited the scope to the local and ground positions because these two positions are involved with aircraft and vehicles on the airport surface during safety-critical operations. The working group also agreed to use Boston-Logan Airport (BOS) as a template to build upon because it contains both parallel and crossing runways.

The working group then implemented a modified version of The Bridge methodology by Dayton, McFarland, & Kramer [5] for structured user interface design. We examined the existing task flows of Alexander et al. [6] and determined that these task flows are still highly accurate. We focused on the most common controller tasks. For the local controller position, we examined 4 of the 7 primary tasks and 24 of the 28 subtasks. For the ground controller position, we examined 3 of the 6 primary tasks and 10 of the 17 subtasks (see Truitt [3], for a complete listing of the tasks we examined). We used the task flows to identify all of the task objects and their respective attributes. We also determined the common actions that controllers would need to act upon each of the task objects.

PROTOTYPE DEVELOPMENT

The task flows, task objects, object attributes, and actions provided the foundation of information requirements necessary to build a GUI. One possible approach was to design an EFDI that preserved the prominent features of a FPS such as the size and information format. Preserving the FPS in an electronic format is reasonable because it seeks to take advantage of expertise that controllers have already developed. However, according to Besnard and Cacitti [7], while the positive transfer of skill is desirable, the preservation of old features can lead to the negative transfer of skill when those features do not work as the user expects.

An alternative approach is to redesign the task altogether. While this approach requires the most change and does not allow much opportunity for the positive transfer of skill, it does present an opportunity to create new, possibly more efficient, methods of operation. Furthermore, relieving the design of historical precedents also promotes the ability to integrate flight data with other information in the ATCT.

To design the GUIs, we formed an interface design team consisting of two ATCT subject matter experts (SMEs), a lead software developer, and an engineering research psychologist. The interface design team used the materials generated by the working group, including the lists of task objects and attributes for each controller position and the airport layout and runway configuration.

ELECTRONIC FLIGHT DATA INTERFACE HARDWARE

The first step of designing the EFDI GUIs was to select the interface hardware. This decision determined how much area would be available to display both flight data and surface surveillance information. The team selected a VarTech Systems, Inc. touch sensitive display. The 21.3" display has an active display area of 17" wide and 12.75" high with a 1600 x 1200 pixel format. The display uses resistive technology to enable a touch screen that can be activated by either a stylus or by fingertip. We mounted the 30.4 lb display on a stand that allows the user to adjust the horizontal and vertical viewing angle. We also acquired Airport Surface Detection

Equipment – Model X (ASDE-X) keyboards and a trackball/keypads as additional input devices (see Figure 1).



Figure 1. Hardware used to implement the prototype EFDIs including resistive touch display, display mount, keyboard, and trackball/keypad.

INTEGRATED ELECTRONIC FLIGHT DATA INTERFACE

An initial suggestion for consolidating information displays came from the Air Traffic Terminal Enhancement and Modernization (ATEAM) group, a group of ATC experts who obtain user input on display integration. Integrating EFD with the ASDE-X display or other surface surveillance system is beneficial in that it places flight data closer to the aircraft positions. Instead of controllers (1) examining FPSs for flight data; (2) shifting their visual attention to the ASDE-X and out-the-window to verify aircraft position; and then (3) mentally correlating these disparate sources of information, the integration of EFD and surface surveillance allows the controller to gain both flight data and aircraft position information from a single source. Integrating information should improve controller efficiency by reducing the controllers' cognitive task load and need to shift visual attention.

The Integrated EFDI contains separate interfaces for the ground and the local controller positions based on the respective tasks, task objects, and attributes identified by the working group (see Figure 2). The primary difference between the EFDIs is the type of lists that appear.

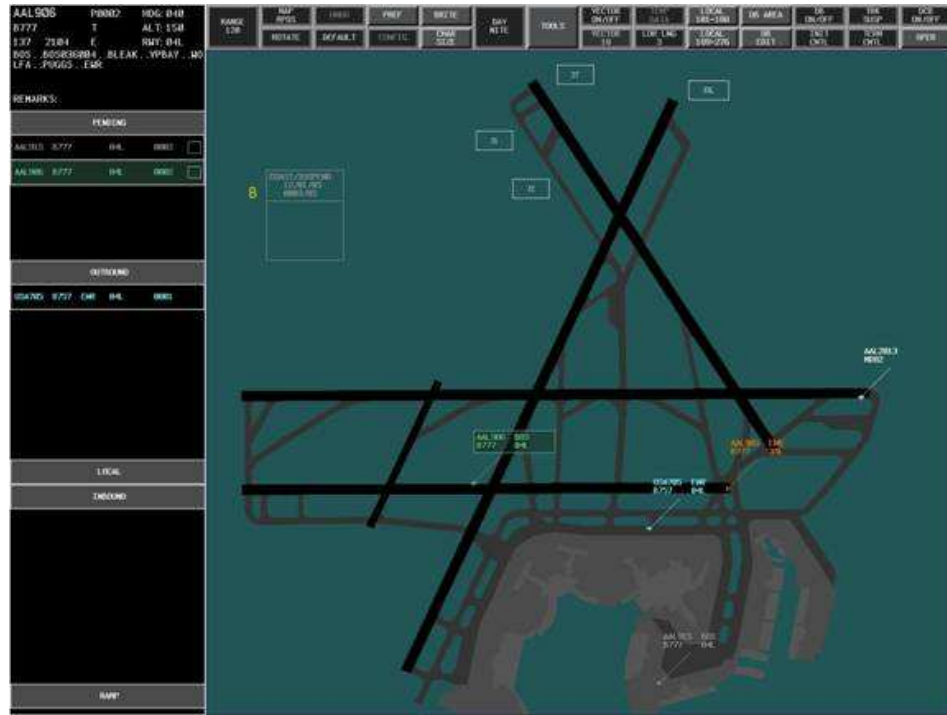


Figure 2. The primary elements of the Integrated EFDI including the situation display, EFD lists, and readout area.

ELECTRONIC FLIGHT DATA LISTS

The EFD lists are similar to FPS bays in that they contain individual flight data elements (FDEs) that can be stacked and sorted. Each FDE contains only the most relevant information, as identified by earlier design processes. Each list has a header that identifies the contents of the list. The list headers are also touch sensitive buttons that allow the user to move FDEs between lists, controller positions, and other facilities by selecting the FDE to move and then selecting the appropriate list header. The user can also resequence FDEs in a list by selecting and dragging an FDE. Selecting either an FDE or data block will cause all other related flight data to display highlighted in green (see Figure 3).



Figure 3. A selected FDE in an EFD list and the selected data block for the same aircraft.

GROUND CONTROLLER LISTS

The pending, outbound, and inbound lists appear on the ground controller's EFDI (see Figure 4). The flight data attributes that appear in the pending list are call sign, aircraft type, runway assignment, proposed departure time or expected departure clearance time (EDCT), and airport traffic information service (ATIS) update indicator. The FDEs in the pending list appear muted (gray text) to indicate that these aircraft have not contacted the ground controller yet. When the controller issues a taxi clearance, she moves the FDE into the outbound list by selecting the aircraft's FDE or data block and then selecting the outbound list header.

PENDING						OUTBOUND						INBOUND	
AAL555	B757	04L	0000			DAL195	B757	EMR	04L	0017		UAL6003	B737
AAL446	B777	04L	0000			USA306	B757	BOS	04L	0017		DAL2153	DC10
DAL195	B757	04L	0000			AAL445	B777	BOS	04L	0017		DAL2154	DC10
USA306	B757	04L	0000			AAL245	B777	EMR	04L	0016		UAL2773	EA32
						USA305	B757	EMR	04L	0016		AAL7904	B737
						AAL246	B777	BOS	04L	0016		AAL1014	MD82

Figure 4. FDEs in the pending, outbound, and inbound lists on the ground controller's Integrated EFDI.

FDEs appear in their active color (white text) at the top of the outbound list when the ground controller moves them from the pending list. Moving an FDE to the outbound list automatically records a taxi time which replaces the proposed departure time in the FDE. The flight data attributes that appear in the outbound list are call sign, aircraft type, first departure fix, runway assignment, taxi time or EDCT, and ATIS update indicator. When the ground controller is ready to transfer responsibility of an aircraft, the controller selects the aircraft's data block or FDE from the outbound list and then selects the local header to transfer it to the local controller.

When the local controller transfers an FDE to the ground controller, the FDE appears at the top of the inbound list on the ground control EFDI. The flight data attributes contained in the inbound list are call sign and aircraft type. The ground controller will provide the aircraft with a taxi clearance back to the ramp and then select the aircraft's FDE or data block followed by the ramp button. This sequence of actions causes the aircraft's flight data to appear muted. The aircraft's data block and FDE will disappear once the aircraft reaches the ramp and the ground controller transfers the FDE to the ramp position.

LOCAL CONTROLLER LISTS

The departure and arrival lists appear on the local controller's EFDI (see Figure 5). When the ground controller transfers an FDE, it appears at the top of the departure list. The departure list contains the flight data attributes of call sign, aircraft type, first departure fix, runway assignment, taxi clearance time or EDCT, and ATIS update indicator. Once an aircraft begins its takeoff roll, the EFDS automatically replaces the time field in the FDE with a timer that begins

to increment. The controller can use this timer for departure spacing. Once the local controller has instructed the aircraft to contact the departure controller, the local controller then selects the appropriate FDE or data block followed by the TRACON button to transfer the FDE.

The flight data attributes contained in the arrival list are call sign, aircraft type, runway assignment, and ATIS update indicator. The local controller transfers the FDE to the ground controller by selecting the aircraft's FDE or data block, followed by the ground button.

DEPARTURE	ARRIVAL
AAL916 B777 BOS 04L 0006 <input type="checkbox"/>	UAL3314 EA32 08 <input type="checkbox"/>
USA706 B757 BOS 09 0004	UAL3304 B737 04R <input type="checkbox"/>
AAL906 B777 BOS 04L 0003	DAL2294 DC10 08
AAL915 B777 BOS 04L 0002	AAL2004 B737 04R
USA705 B757 EWR 04L 0001	UAL3303 B737 04R
AAL905 B777 EWR 04L 01:27	DAL2293 DC10 04R
	AAL2003 B737 04R

Figure 5. FDEs in the departure arrival lists on the local controller's Integrated EFDI.

READOUT AREA

The readout area is located above the EFD lists. Three different types of information may appear in the readout area: full flight data for an arrival aircraft, full flight data for a departure aircraft, or a list of the most recent FDEs transferred to another controller position or facility. When a controller selects a data block or an FDE, the full set of flight data attributes appears in the readout area (see Figure 7). When the controller selects a departure aircraft's FDE or data block, the readout area displays the aircraft's call sign, type, computer identification, beacon code, proposed departure time, taxi time, EDCT, assigned heading, assigned altitude, assigned runway and intersection departure, full route of flight, and remarks. When a controller selects an arrival aircraft's FDE, the aircraft's call sign, type, computer identification (CID), runway assignment, remarks, and missed approach button appear in the readout area.

AAL906 P0002 HDG: 040 B777 T0004 ALT: 150 137 2104 E RWY: 04L BOS . BOS036004 . BLEAK . VPBAY . W0 LFA . PUGGS . EWR REMARKS:	DAL2293 HDG: DC10 ALT: 129 RWY: 04R REMARKS: <div> MISSED APPROACH </div>
---	---

Figure 7. The readout area information for a departure aircraft (left) and arrival aircraft (right).

The controller can also display in the readout area the last four FDEs transferred to another position or facility by selecting the button for that position or facility. The controller can then recall any of the FDEs displayed in the readout area by selecting the FDE and then selecting a list header to place the FDE at the top of the appropriate list.

The controller can amend flight data by using the touch screen and keyboard. When a controller amends flight data, an asterisk will appear on the right hand side of the aircraft's FDE (see Figure 8) and appears highlighted in the readout area (see Figure 9). When a controller transfers the FDE to another controller, the asterisk notifies the receiving controller that there has been a change. The controller selects the FDE displaying an asterisk to examine the amended flight data in the readout window. The amended flight data appears highlighted until the controller acknowledges them by touching the readout area.



Figure 8. FDE indicating that there has been a change to the flight data..

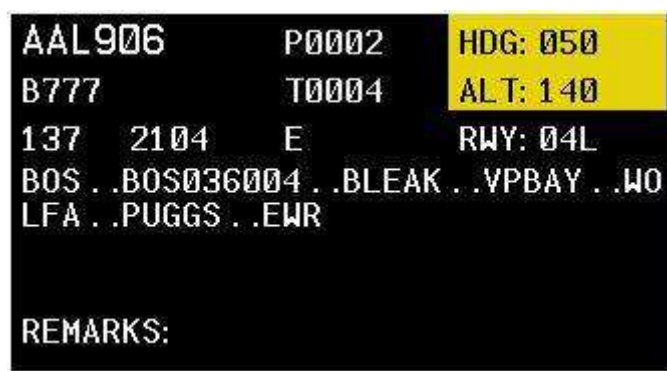


Figure 9. Amended altitude and heading assignments as depicted in the readout area.

DATA BLOCKS

We designed the data blocks to reflect only the most relevant information for the controller (see Figure 6). The departure data block contains four flight data attributes including aircraft call sign, aircraft type, first departure fix, and runway assignment. The arrival data block contains only two flight data attributes: aircraft call sign and aircraft type.



Figure 6. Departure (left) and arrival (right) data blocks.

BUTTONS

Figure 10 shows buttons on the surface situation display that the controller can use to assign and record runway and intersection departure assignments. Both the ground and local controllers can change an aircraft's runway assignment or assign an intersection departure by selecting the aircraft's FDE or data block and then selecting the appropriate button.



Figure 10. Buttons on the Integrated EFDI situation display for runway and intersection departure assignments.

REMINDERS

According to the FAA Order 7110.65P [8], controllers must use a procedure on initial contact with an aircraft to verify that the pilot has the most recent ATIS information. The ATIS update indicator on the Integrated EFDI alerts the controller whenever the ATIS changes. An ATIS change automatically causes the ATIS code to flash near the system information window. The Integrated EFDI also indicates which aircraft to advise of the change. For example, on the ground controller's EFDI, all FDEs appear in the pending list with an ATIS update indicator (box) on the right hand side (see Figure 11). Once the controller provides the current ATIS information to the pilot, the controller touches the indicator in the aircraft's FDE to make it disappear. When the ATIS information updates, all FDEs in the pending, outbound, departure, and arrival lists will display the indicator once again.

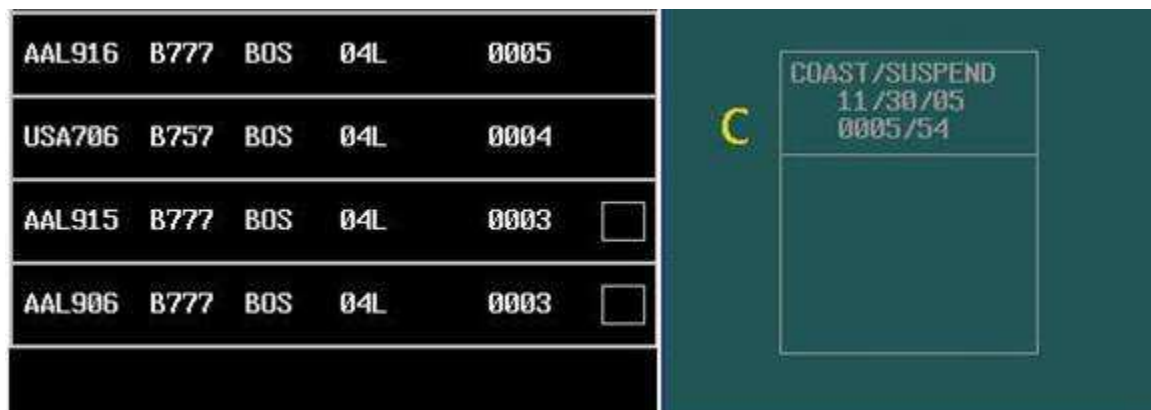


Figure 11. FDEs with ATIS update indicator and ATIS code near system information window.

We designed timers to reduce the controller's cognitive workload. The runway spacing timer works automatically with departure aircraft to assist the controller in determining appropriate

departure spacing. Once an aircraft begins its takeoff roll (as sensed by surface surveillance), the time field in the aircraft's FDE is automatically replaced by a timer that begins to increment from zero. This timer alleviates the controller's need to remember or record the departure time and then mentally calculate the next departure time.

To set the timer generically, the controller selects the current time/date field in the system information window to activate the timer interface (see Figure 12). The controller selects the amount of time desired and then selects the start/stop button to begin a countdown. Once a controller starts a generic timer, the timer appears next to the system information window. When the timer expires it notifies the controller by flashing the expired timer.



Figure 12. The Integrated EFDI Timer Interface.

The controller can also associate a timer with a particular aircraft by selecting the aircraft's FDE or data block and then activating the timer interface. This action places a timer icon in the aircraft's FDE (see Figure 13). When the timer expires, it notifies the controller by flashing the timer icon.



Figure 13. FDE for an aircraft that has an associated timer.

The working group's task analysis and recently collected observational field data by Dattel et al. [9] show that controllers use a number of techniques to make particular flight data more conspicuous. Controllers can highlight flight data by selecting an FDE or data block and then selecting the readout area. This action causes the text in the FDE and data block to appear in light blue (see Figure 14). To remove the highlighting, the controller selects a highlighted FDE or data block and then selects the readout area.

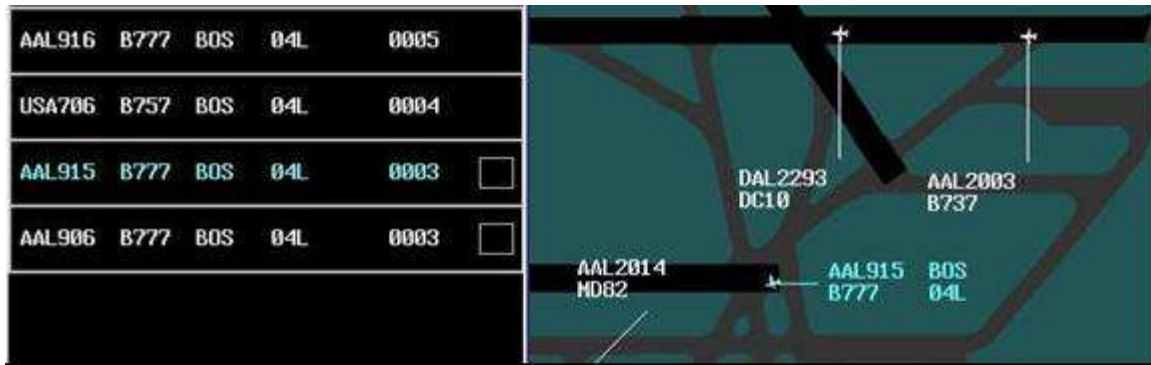


Figure 14. A highlighted FDE and the associated data block.

Aircraft waiting on an active runway present a potential safety risk that is inherent in the procedure known as taxi-into-position-and-hold (TIPH). Controllers must remember when an aircraft is holding on the runway to prevent possible runway incursions and collisions. The Integrated EFDI uses the capabilities of the surface surveillance to automatically detect and indicate when an aircraft is holding or stopped on an active runway by highlighting the aircraft's FDE and data block in orange (see Figure 15). Once the aircraft begins its takeoff roll, the Integrated EFDI automatically removes the highlighting.

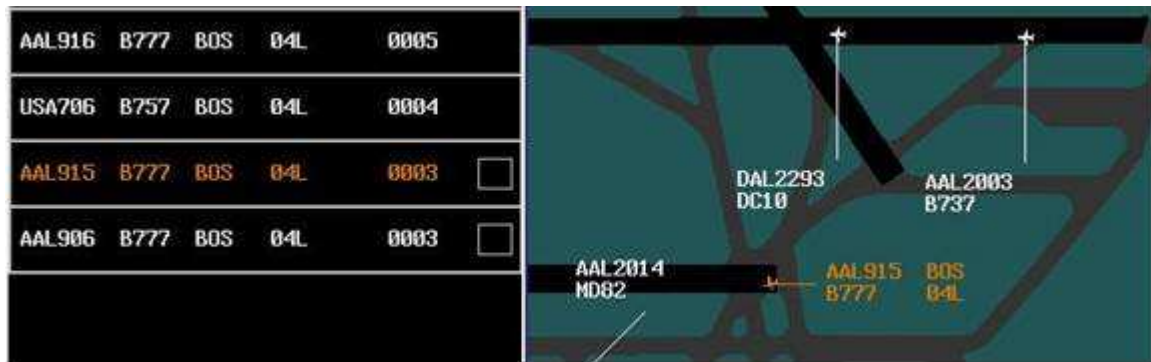


Figure 15. Highlighted aircraft that is holding on the runway.

DATA BLOCK MANAGEMENT

Our design reduces inadvertent selections by automatically eliminating clutter from the display and reducing the chance that two elements occupy the same location on the display. When an aircraft first appears on the situation display, the Integrated EFDI automatically orients the leader line based on the aircraft's runway assignment. An automatic data block management algorithm also separates the data blocks as they traverse the display. Leader line positions may occupy any orientation. Automatic data block management should reduce controller workload and help ensure that information is available to the controller at all times. Controllers can also select and drag a data block to move it to a new position.

PERCEPTUAL-SPATIAL ELECTRONIC FLIGHT DATA INTERFACE

We designed the Perceptual-Spatial (P-S) EFDI as an alternative for ATCTs that do not have surface surveillance capability (see Figure 16). The P-S EFDI also serves as a backup system for the Integrated EFDI in case surface surveillance fails. The P-S EFDI shares most of the features

of the Integrated EFDI, but presents them in a fundamentally different way. This design overview describes the unique aspects of the P-S EFDI.

The fundamental concept of the P-S EFDI is to reduce controller physical and cognitive workload by providing a means to track, record, communicate, and organize EFD spatially in relation to the airport surface. We designed the P-S EFDI to help controllers correlate flight data more easily with the aircraft they represent. Controllers could use this tool in conjunction with a surface surveillance display. The P-S EFDI uses the same hardware as the Integrated EFDI.



Figure 16. The primary elements of the P-S EFDI including FDEs, the readout area, buttons, and system information window.

FLIGHT DATA ELEMENTS

The P-S EFDI does not contain data blocks, but the FDEs are similar in form to that of a data block. The FDEs occupy space on the airport surface map and the controller can visually categorize them by their color, shape, location, and flight data attributes. All of the EFD functions operate in a manner similar to the Integrated EFDI. However, with the P-S EFDI, the controller must move each FDE and place it on the airport surface map. Placing an FDE in certain areas of the map has different effects including time records and highlighting.

We designed the FDEs to minimize their size while still maintaining usability. Along with the other primary elements, the outbound, inbound, and arrival FDEs appear on both the ground and local controller positions. The pending FDEs only appear on the ground controller's position; the departure FDEs only appear on the local controller's position. FDEs that are in the

possession of a controller appear with white text, while pending FDEs, or FDEs possessed by another controller, appear with gray text. Except for pending FDEs on the ground controller's interface, a controller can only move FDEs that are in her possession. This possession rule preserves the FDEs' usefulness for communicating information about aircraft location and prevents controllers from interfering with each other's actions.

GROUND CONTROLLER FLIGHT DATA ELEMENTS

Pending, outbound, and inbound FDEs appear on the ground controller's EFDI (see Figure 17). A pending FDE contains aircraft call sign, type, first departure fix, runway/intersection assignment, proposed departure time or EDCT, and an ATIS update indicator. The pending FDEs appear on their designated ramp spot with gray text to indicate that the ground controller has not yet contacted the aircraft.



Figure 17. Pending, outbound, and inbound FDEs on the ground control P-S EFDI.

An outbound FDE contains the flight data attributes of call sign, aircraft type, first departure fix, runway/intersection assignment, assigned taxi time or EDCT, ATIS update indicator, and flight data update indicator or timer indicator. When the ground controller moves an FDE out of the ramp area onto the airport surface map it becomes an outbound FDE, a taxi time is automatically assigned, and text changes from gray to white. To transfer the outbound FDE to the local controller, the ground controller selects the FDE and then selects the local controller button. This causes the FDE on the ground controller's interface to display with gray text and the associated FDE on the local controller's departure list will display with white text.

Once the local controller transfers an arrival FDE to the ground controller, the FDE becomes an inbound FDE. An inbound FDE shows the flight data attributes of aircraft call sign and type and appears with white text on the ground controller's EFDI and with gray text on the local controller's EFDI. The ground controller transfers inbound FDEs to a ramp controller or AOC by selecting an FDE and then selecting the ramp button.

LOCAL CONTROLLER FLIGHT DATA ELEMENTS

Outbound, departure, arrival, and inbound FDEs appear on the local controller's EFDI. The controller can indicate a TIPH clearance by selecting an outbound FDE and then selecting the appropriate TIPH button. This causes the FDE to occupy the location of the TIPH button and to display with orange text (see Figure 18).



Figure 18. Runway spacing timers, departure clearance button, and occupied and unoccupied TIPH buttons on the local controller's EFDI.

When the controller clears an aircraft for departure, the controller selects the FDE and then selects the departure clearance button (see Figure 18). Assigning a departure clearance automatically assigns a departure time for that aircraft and the FDE occupies the departure list beneath the appropriate runway assignment button (see Figure 19). Simultaneously, the departure FDE displays a departure spacing timer in the FDE time field. The runway spacing timer also appears above the TIPH button for the appropriate runway to indicate the time since the last aircraft departed from that runway. The controller can transfer a departure FDE to the TRACON by selecting an FDE and then selecting the departure header in the departure list. This causes the FDE to disappear. The runway spacing timer resets and begins counting up from zero again when the local controller clears the next aircraft to depart from the same runway.



Figure 19. FDEs for aircraft that the local controller has cleared for departure appearing in the departure list on the local controller's EFDI.

Arrival FDEs enter at the top of the arrival list and are ordered by time sequence over the outer markers (see Figure 20). The arrival FDEs contain aircraft call sign, aircraft type, runway assignment, and ATIS update indicator. The FDEs for aircraft landing on different runways automatically offset from one another in the arrival list. The controller can resequence the arrival FDEs by selecting an FDE and dragging it to a new location within the arrival list. Once an aircraft has landed, the local controller can drag the aircraft's FDE to the appropriate location on the airport surface map to indicate its approximate location. The local controller transfers control of an FDE to the ground controller by selecting the FDE and then selecting the ground button.

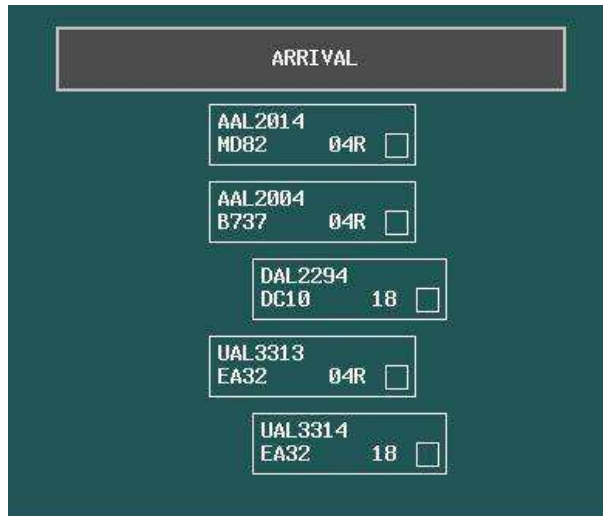


Figure 20. Arrival FDEs in the arrival list on the local control P-S EFDI.

FLIGHT DATA ELEMENT CHAINS

We discovered a number of usability issues related to FDE management during initial, low-risk usability testing and rapid prototyping. One issue we identified relates to sequences, or stacks, of FDEs positioned on the airport surface map. The outbound and departure FDEs are likely to stack up while aircraft are waiting to cross an active runway or waiting to depart at the end of a runway. To maintain stacks of FDEs, we designed three zones on the airport surface map where FDEs link together in a chain. The zones are not visible to the controller. When the controller moves an FDE into one of the zones, the FDE displays a stem on top of the FDE to indicate that it is in a zone. The FDE becomes part of a chain when the controller releases it in a zone (see Figure 21). The first FDE in a chain occupies an anchor position at the top of the zone. When the controller moves an FDE that is part of a chain, the remaining FDEs in the chain automatically move together and the FDE at the front of the chain occupies the anchor position.



Figure 21. Three chains of FDEs on the local control P-S EFDI.

SUMMARY AND CONCLUSION

We developed both the Integrated and P-S EFDI prototypes based on the same foundation of information gleaned from the existing literature, task analyses, and SME opinion. Despite their common foundation, each prototype EFDI provides controllers with fundamentally different ways to manage EFD. In the near future, we will submit each of the prototype EFDIs to rigorous usability testing in the laboratory. We will modify each of the prototype EFDIs based on the results of such testing. Once we have established their usability in low-to-medium fidelity simulations, we recommend continued testing in more complex high fidelity simulations that will empirically test the prototype EFDIs, compare them to FPSs, and assess their effect on controller workload, communication, situation awareness, and overall performance. A provisional patent application is pending for the EFDIs.

REFERENCES

1. Federal Aviation Administration. (2005, August). *Administrator's fact book*. Washington, DC: Author.
2. Durso, F. T., & Manning, C. A. (2002). Spinning paper into glass: Transforming flight progress strips. *Human Factors and Aerospace Safety*, 2(1), 1-31.
3. Truitt, T. R. (2006). *Concept Development and Design Description of Electronic Flight Data Interfaces for Airport Traffic Control Towers* (DOT/FAA/TC-TN-06/17). Atlantic City, NJ: U.S. Department of Transportation, Federal Aviation Administration.
4. Truitt, T. R. (2006). *Electronic flight data in airport traffic control towers: Literature review* (DOT/FAA/CT-05/13). Atlantic City, NJ: U.S. Department of Transportation, Federal Aviation Administration.
5. Dayton, T., McFarland, A., & Kramer, J. (1998). Bridging user needs to object oriented GUI prototype via task object design. In L. E. Wood (Ed.), *User Interface Design*. CRC Press.
6. Alexander, J. R., Alley, V. L., Ammerman, H. L., Fairhurst, W. S., Hostetler, C. M., & Jones, G. W., et al. (1989). *FAA air traffic control operations concepts. Vol. 7: ATC tower controllers* (FAA Report No. DOT/FAA/AP-87-01). Washington, DC: Federal Aviation Administration.
7. Besnard, D., & Cacitti, L. (2005). Interface changes causing accidents: An empirical study of negative transfer. *International Journal of Human-Computer Studies*, 62, 105-125.
8. Federal Aviation Administration. (2004). *Air Traffic Control* (DOT/FAA Order 7110.65). Washington, DC: Author.
9. Dattel, A. R., Johnson, B. R., Durso, F. T., Hackworth, C., & Manning, C. A. (2005). Flight progress strips in towers: Frequency index and perceived psychological benefits. In *Proceedings of the 13th International Symposium on Aviation Psychology*. Oklahoma City, OK.